REMARKS/ARGUMENTS

The final office action mailed on October 26, 2010, has been reviewed and carefully considered. Reconsideration is respectfully requested.

Amendments to the Claims

Claims 1-6 were pending in the present application prior to this amendment. Claims 1, 4-6 and 10-11 are now pending in the present application; among them, claim 1 is the only independent claim. Claims 1 and 4 have been amended, where claim 1 has been amended to include the limitations of claims 2-3. New dependent claims 10-11 have been added, which also depend from claim 1. Claims 2-3 have been canceled without prejudice. No new matter has been added.

Claim Rejections - 35 U.S.C. §103

In the office action (page 3), claims 1-2 stand rejected under 35 U.S.C. §103(a) as being obvious over U.S. Patent No. 7,013,424 (James) in view of U.S. Patent No. 7,305,615 (Ross).

In the office action (page 5), claims 3-4 stand rejected under 35 U.S.C. §103(a) as being obvious over James in view of Ross and U.S. Patent No. 6,938,204 (Hind).

In the office action (page 7), claims 5-6 stand rejected under 35 U.S.C. §103(a) as being obvious over James in view of Ross and U.S. Patent No. 7,080,094 (Dapp).

Comments for 103 Rejections

The present invention relates to XML (extensible mark-up language) processing by parsing, and more particularly, to a separate and independent XML processor not part of the conventional software parsing of the received XML document **in which a part of XML processing performed by the conventional software parsing on a received XML documents is now performed by a hardware to not only reduce load**

of system but also improving an XML processing speed performed by the conventional software parsing on a received XML documents.

Also, the presently claimed invention relates to an XML processing method in a system having the separate and independent XML processor from the software XML for processing a received XML document, **while this independent XML processor produces the same output on the received XML document as if the XML processing was done by software on the same XML document**. As shown in FIG. 1 of the presently claimed invention, FIG. 1 discloses is a schematic diagram illustrating operations of an XML processor and conventional generalized XML parser based on software of a received XML documemt. As shown in FIG. 1, an XML processor 13 performed by the hardware according to the presently claimed invention and software-based conventional generalized XML parsers 11 not performed by hardware are **both able to receive and process the same input XML document 10 and generate the same product 12, where when the processing of the receive input XML document by XML processor 13, the XML processing time due to parsing is reduced over the software-based conventional generalized XML parsers 11 not having the claimed hardware**.

As a result, the hardware based XML processor 13 of the presently claimed invention reduces the processing time of the software based XML processor for an XML document through a tree structure by using a node usage check table 301 and a node table 302 and storing data of a real node in a node memory 303, which does **not** increase a size of a memory and navigates the tree structure in an application program at a high speed. **More specifically, the application program of the presently claimed invention promptly confirms whether the node usage check table includes a node table corresponding to a desired node, and access a memory in which data of the desired node is stored through the node table**.

Further, the presently claimed invention recites displaying an index indicating whether a node table corresponding to each field of the node usage check table is used, and storing an address of a memory in each field of the node table, which does not require an excessively large sized memory although the memory has a fixed size.

Nowhere in James able to disclose that the same XML document is received by the special purpose processor and the general purpose processor because the special purpose processor of James in only used to **offload** the processing done by the general purpose processor (James Abstract; col. 5, lines 5-22; and FIG. 3).

Accordingly, how can James have the general purpose processor receive a same input such that the hardware and the software portion of the XML processing produces the same output if general purpose processor performing the software processing the XML documents only receives the XML document **after** the special purpose processor has offloaded the part of the processing by the gateway 346 as shown in FIG. 3 of James. James confirms that the process performed by the gateway 346 is not the same document performed by the software of the general purpose processors 310 a-e because the general purpose processor is only capable of processing HTML and the special purpose processor is only capable of processing XML (James page 7, lines 1-25 and FIG. 3)?

Generally, the prior art software-based conventional generalized XML parsers are viewed as a software library used to facilitate manipulation of XML documents. Most conventional XML parsers are configured to be compatible with XML grammar. However, a significant drawback of the conventional XML parsers is that such conventional parsers require relatively large software components, which causes load of a system that processes the increase uses from receiving XML documents through the internet, where the Internet has increased the receipt of XML document due to increase of data from the information-oriented era. In particular, usage of web has been rapidly increasing in a variety of embedded systems, such as cellular phones, digital home electronics, telematics terminals, PDAs (Personal Digital Assistant), web TVs, and the like, besides typical PCs. However, these embedded systems typically have limited computing power and memory capacity, when being compared to say for example a PC. As a result, the software-based conventional XML parsers of the prior art are generally not suitable for use in embedded systems.

In contrast, the presently claimed invention discloses an **efficient hardware based XML parsers for reducing the time required to parse XML documents previously performed through programming by promptly confirming whether the**

node usage check table includes a node table corresponding to a desired node, and access a memory in which data of the desired node is stored through the node table. Thus, the presently claimed invention is better suited for these non-PC-based devices in order to reduce the loads put on these non-PC based devices when performing XML processing in theses non-PC-based devices.

Accordingly, the presently claimed invention provides an XML processor in which a part of software-based XML processing performed on the received XML document is performed in a hardware manner based on independent hardware, thereby improving an XML processing speed previously done by conventional software-based processing/parsing of a received XML document(s).

More specifically as stated above, the hardware based XML processor 13 of the presently claimed invention reduces the processing time of the software based XML processor for an XML document through a tree structure by using a node usage check table 301 and a node table 302 and storing data of a real node in a node memory 303, which does **not** increase a size of a memory and navigates the tree structure in an application program at a high speed. More specifically, the application program of the presently claimed invention promptly confirms whether the node usage check table includes a node table corresponding to a desired node, and access a memory in which data of the desired node is stored through the node table.

Further, the presently claimed invention recites displaying an index indicating whether a node table corresponding to each field of the node usage check table is used, and storing an address of a memory in each field of the node table, which does not require an excessively large sized memory although the memory has a fixed size.

Claim 1 has been amended to <u>better</u> clarify this above described novel aspect of the presently claimed invention, which recites inter alia:

> --a second memory employed by the hardware processing module; and
>
> a CPU controlling the XML processing on the received XML document by the software stored in the first memory to generate a first output if the XML is executed by software, and to generate a second output if the part of the XML processing is performed in the hardware manner,
>
> **<u>wherein the second memory comprises:</u>**
>
> **<u>a node memory storing the whole information that each node</u>**

has to store, at least one of a node name, a node type, a parent node, a child node;
a node table managing the information stored in the node memory;
a node usage check table indicating whether to use the node table,
wherein the hardware processing module assigns storage to each node, re-assigns the storage in each node and returns the storage in each node, using the node usage check table
~~wherein the XML processing time is reduced from the hardware processing module performing the part of the XML processing in the hardware manner, and~~
~~wherein the first and second outputs are equivalent~~--.

The Applicants respectfully submit that the cited references do not describe, teach, or suggest each and every one of the limitations recited in amended clam 1. That is, nowhere in James nor Ross nor Dapp, neither alone or in combination, discloses or suggests relates to XML (extensible mark-up language) processing by parsing, and more particularly, to a separate and independent XML processor not part of the conventional software-based parsing of the received XML document in which a part of XML software-based processing preformed on the received XML document is performed by a hardware, which if was performed in either hardware or software would produce the same output, by having a second memory associated with the XML processing performed by hardware in order to produce the same output as if performed by software, which this second memory includes: a node memory storing the whole information that each node has to store, at least one of a node name, a node type, a parent node, a child node; a node table managing the information stored in the node memory; a node usage check table indicating whether to use the node table such that **the hardware processing module assigns storage to each node, re-assigns the storage in each node and returns the storage in each node, using the node usage check table**.

Accordingly, the applicants respectfully submit that nowhere are the cited reference are able reduce load of system through the second memory of hardware processing of the XML document, which in turn also improve the XML processing speed, wherein the generated output performed by the second memory in hardware on the received XML documents produces a same output as if the part of the XML processing performed in the hardware manner.

In contrast, James discloses using special purpose processor implemented in hardware, which is in front of the general purpose processor, to "offload" processing done by the general purpose processor (OA page 9 under "Response to Arguments". Accordingly, nowhere can James disclose receiving an XML document and then having both the general purpose processor(s) (310 a-e) and the special purpose processor produce the same outputs from parsing the received XML document because the input to the general purpose processor is **different** (i.e.; as understood by the examiner, see OA page 9) from the input to the special purpose processor (346) (James FIG. 3). Ross, Dapp, and Hind fail to make for the deficiency of James.

In contradistinction, FIG. 4 of the presently claimed invention illustrates an XML processor performing part of the software-based XML processing/parsing by hardware as follows:

> "An XML processor 13 shown in FIG. 4 comprises a CPU 40 that generally controls the XML processor 13, a memory 41 that stores software for performing a specific function of the CPU 40, variables, and values required to execute software instructions, a hardware processing **module 42 that performs a part of XML processing in a hardware manner**, and a memory 43 used in the hardware processing module 42. A bus 44 that receives and transmits data connects the above components.
>
> FIG. 4 shows that a specific function among the XML processing functions can be realized in **a hardware manner**. For example, a memory management function used in parsing, i.e., processing of assigning, returning, and reassigning memory, influences the most the performance of software parsers.
>
> Referring to FIG. 4, **the XML processor according to the embodiment of the present invention can realize the memory management function in a hardware manner in order to improve the performance of XML processing**. The XML processor according to the embodiment of the present invention can realize an XML DTD and a state machine with respect to an XML schema, which are frequently used in XML processing, in a hardware manner, in addition to the memory management function",

(specification page 4, lines 6-23 and FIGs. 4-8 [**emphasis** added]).

Further in contradistinction, FIG. 5 of the presently claimed invention shows a schematic diagram illustrating a method for realizing a memory management function in a hardware manner with respect to the XML processor 13 of FIG. 4, where XML parsers

compile an **XML document into a tree structure, providing hierarchical information with which application programs are able to navigate** as follows.

> "XML elements are expressed as nodes, which are **correlated and tree-based**. Assignment, reassignment, and return of memory have to be processed for the nodes. What are considered for memory management with respect to the node are to select information to be maintained at the node and to process information having flexible data size.
>
> Information that the nodes have to store may be defined according to DOM (Document Object Model), one of W3C standards for XML document processing. DOM is a standard supported by most XML parsers so that it can be easily linked with the existing parsers of various types.
>
> Referring to DOM, information that the nodes have to store includes a node name, a node value, a node type, a parent node, a child node, an eldest child node, a youngest childe node, a preceding sibling, a following sibling, and an attribute value. More information, in this regard is presented in DOM standards.
>
> The information having flexible data size is a node value when a node type is PCDATA. Sizes of a node name, a node type, and an attribute value having a data value are hardly so flexible that there are few problems even if sizes of them are processed as being fixed. Other information is pointer information and thus is enough to store only address information.
>
> In order to satisfy the above characteristics, the memory management function using hardware may be composed of a node usage check table 301, a node table 302, and a memory 303 that maintains actual data, as shown in FIG. 5. The node usage check table 301 is divided into several blocks having different sizes depending on the data size that the nodes have to store. Fields in the blocks indicate whether to use the corresponding node table 302. The node table 302 manages the whole information that the nodes have to store, i.e., a node name, a node type, a parent node, a child node, and the like. **However, the node table 302 stores not actual values but addresses of the memory 303. Every field of the node table 302 has a fixed size**.
>
> The memory 303 may be RAM (Random Access Memory) and is used for storing actual values. It is possible to realize the node usage check table 301 and the node table 302 as an ASIC (Application-Specific Integrated Circuit).
>
> The node usage check table 301 and the node table 302 shown in FIG. 5 have a fixed correlation. To be more specific, if locations of a memory to be used in the node usage check table 301 are determined, locations of a block to be used in the node table 302 are automatically determined. **The node table 302 does not store actual values but stores addresses of the memory 303 having actual information**. The

> node table 302 may be mapped to fixed locations in order to obtain maximum performance",

(specification page 5, line 28 to page 6, line 33 and FIG. 5 [**emphasis** added])

Accordingly, the presently claimed invention discloses that there is little information having actual values and that there is little information having large variability on the node table 302. Therefore, if the memory of the presently claimed invention has a proper size, **memory will not be greatly consumed**. The addresses to be stored in the node table 302 may be information of other fields (i.e., other nodes) in the node table 302.

As described above, the presently claimed invention provides an XML processor in which a part of software-based XML processing/parsing is performed on a received XML document in a hardware manner based on independent hardware, which is separate from the software-based processing of the XML document but produces the same output from the software-based processing of the XML document, and which means that part of the software-based processing is now being performed by hardware to generate a same output as if the received XML document was completely processed by the conventional software-based XML processing. Thereby, the software with part of the processing of an XML document by hardware disclosed by presently claimed invention improves an XML processing speed done previously only by software and thus reduces computational load (i.e.; eliminates software processing of an XML document) of a system for using the convention software-based XML processing on a received XML document.

Therefore, the applicants respectfully submit that nowhere does James nor Ross nor Dapp nor Hind nor any of the examiner's cited references, neither alone nor in combination, disclose nor suggest each and every one of the limitations recited in amended claim 1 above.

DEPENDENT CLAIMS

The other claims are dependent from independent claim 1 discussed. Thus, the remaining dependent claims are therefore believed patentable for at least the same reasons. Since each dependent claim is also deemed to define an additional aspect of

the invention, however, the individual reconsideration of the patentability of each on its own merits is respectfully requested.

Conclusion

For the reasons set forth above, the applicants respectfully submit that claims 1, 4-6 and 10-11, now pending in this application, are in condition for allowance over the cited references. Accordingly, the applicants respectfully request reconsideration and withdrawal of the outstanding rejections and earnestly solicit an indication of allowable subject matter.

This amendment is considered to be responsive to all points raised in the office action. The examiner is encouraged to contact the undersigned attorney by telephone to expeditiously resolve any remaining questions or concerns.

Respectfully submitted,

Dated: January 4, 2011

Keith S. Van Duyne, Reg. No. 54,505
Ladas & Parry LLP
224 South Michigan Avenue
Suite 1600
Chicago, Illinois 60604
(312) 427-1300